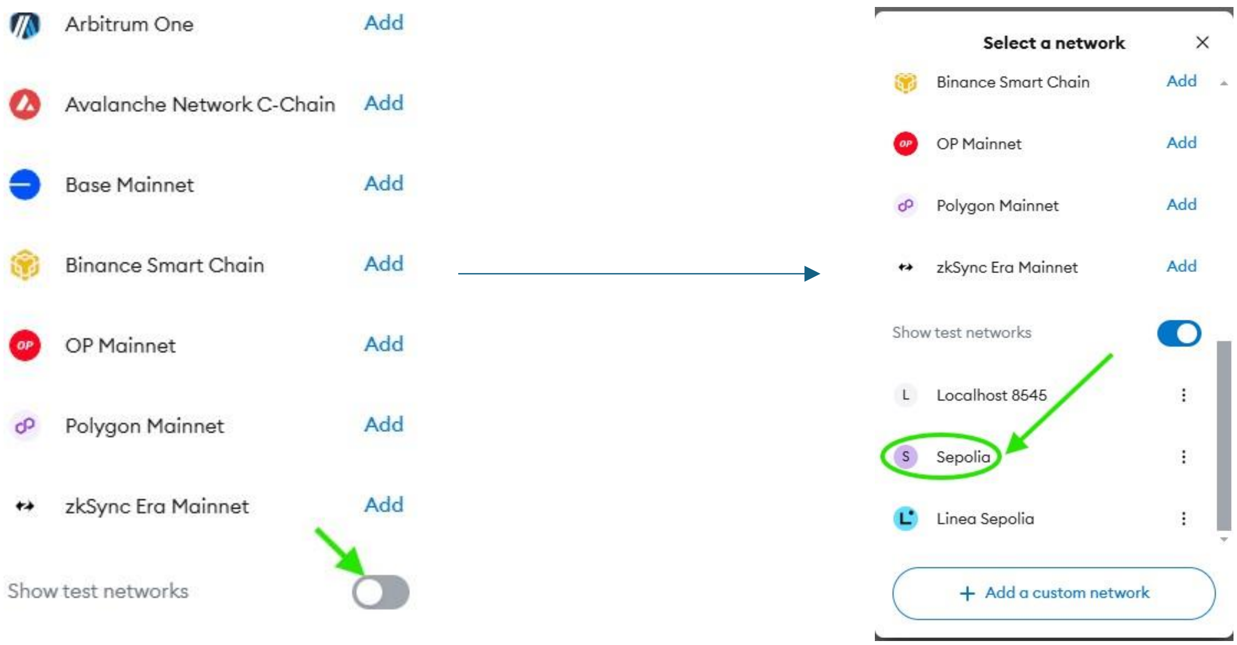
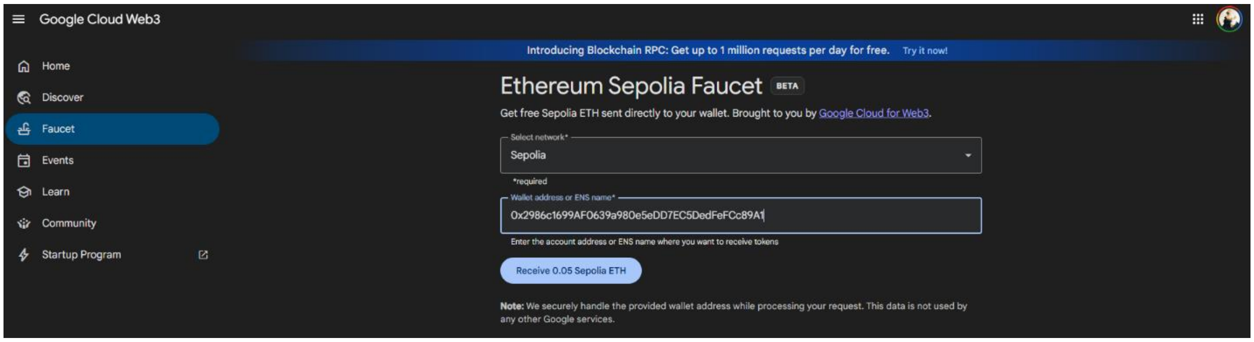
Change the network to **Sepolia Testnet :**



Go to https://cloud.google.com/application/web3/faucet/ethereum/sepolia :



Introducing Blockchain RPC: Get up to 1 million requests per day for free. Try it now!

Ethereum Sepolia Faucet BETA

Get free Sepolia ETH sent directly to your wallet. Brought to you by Google Cloud for Web3.

Select network*: Sepolia

*required

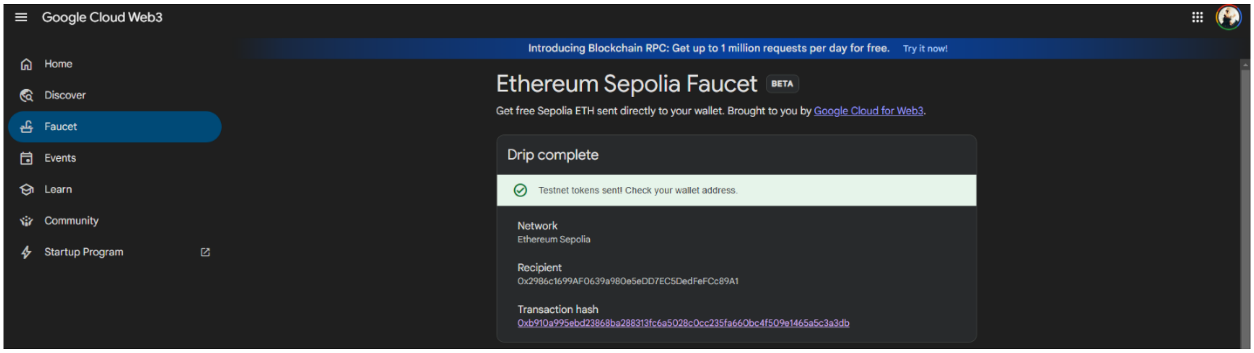
Wallet address or ENS name*: 0x2986c1699AF0639a980e5eDD7EC5DedFeFCc89A1

Enter the account address or ENS name where you want to receive tokens

Receive 0.05 Sepolia ETH

Note: We securely handle the provided wallet address while processing your request. This data is not used by any other Google services.

Paste the wallet address and receive **0.05 ETH per Day**:



Introducing Blockchain RPC: Get up to 1 million requests per day for free. Try it now!

Ethereum Sepolia Faucet BETA

Get free Sepolia ETH sent directly to your wallet. Brought to you by Google Cloud for Web3.

Drip complete

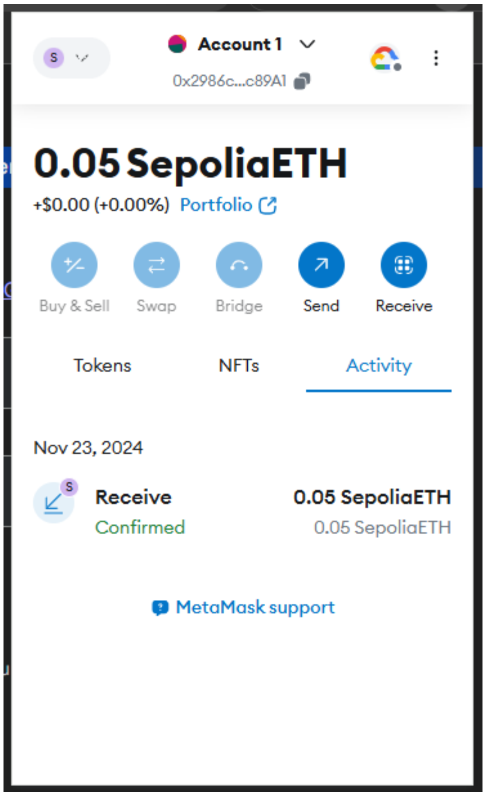
✓ Testnet tokens sent! Check your wallet address.

Network
Ethereum Sepolia

Recipient
0x2986c1699AF0639a980e5eDD7EC5DedFeFCc89A1

Transaction hash
0xb910a995ebd23868ba288313fc6a5028c0cc235fa660bc4f509e1465a5c3a3db

We should see the **0.05 ETH** deposited into our wallet address:



0x2986c...c89A1

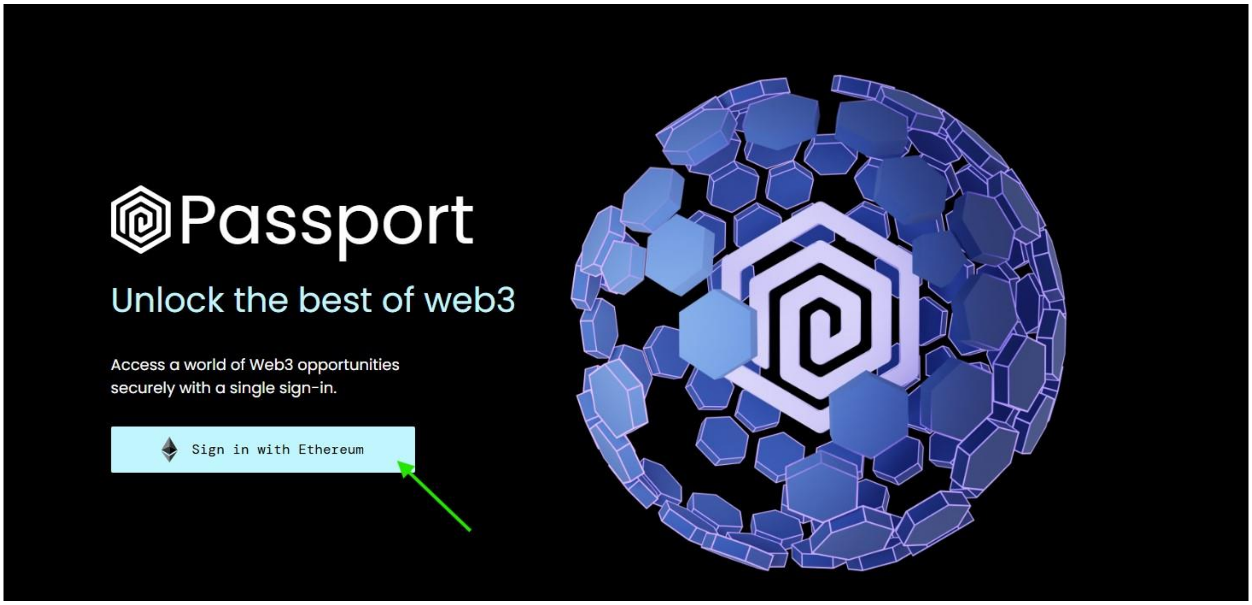
0.05 SepoliaETH

+$0.00 (+0.00%) Portfolio

Nov 23, 2024

Receive — Confirmed — 0.05 SepoliaETH — 0.05 SepoliaETH

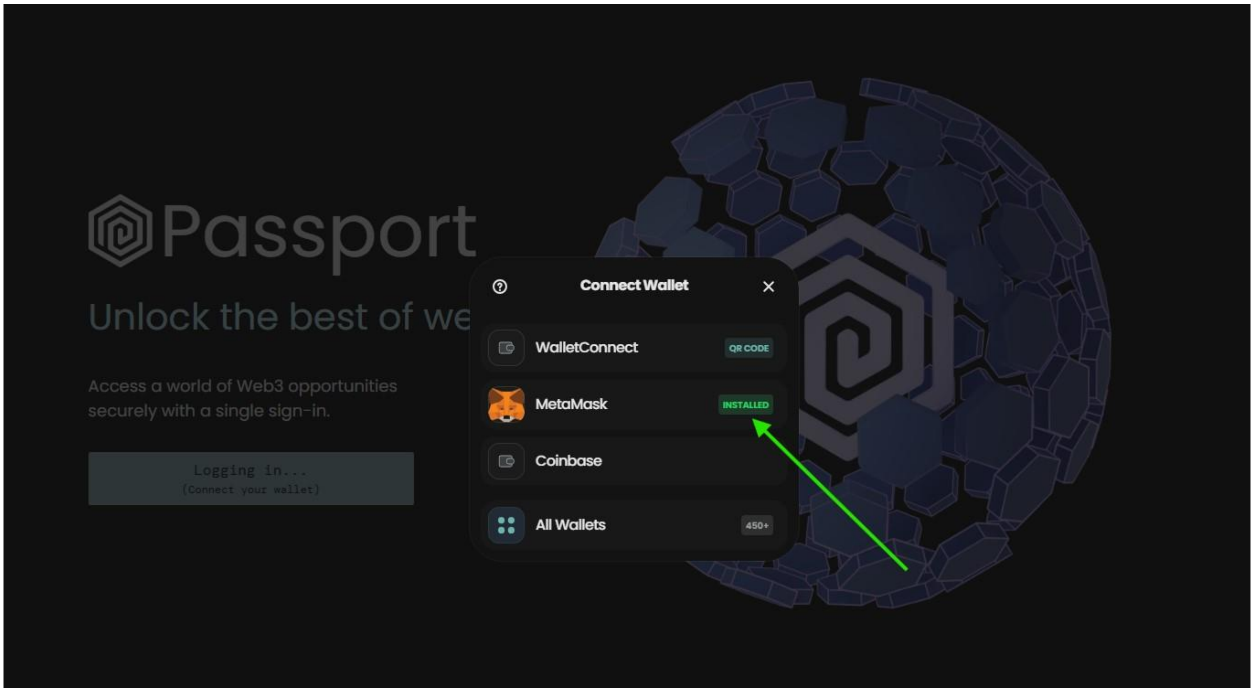
Another way to get more ETH is mining testnet ETH, if we're developing a complex program. This part is optional. Go to https://app.passport.xyz/#/ :



Unlock the best of web3

Access a world of Web3 opportunities securely with a single sign-in.

Sign in with Ethereum using **MetaMask**:

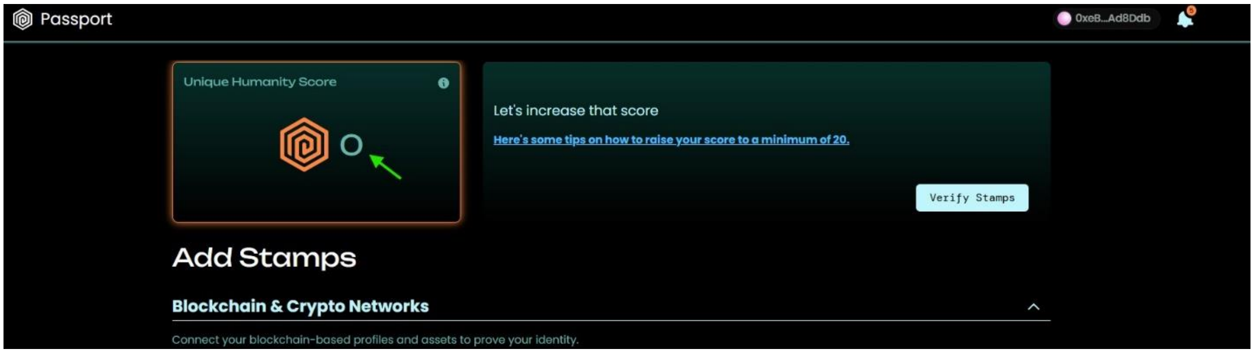


Passport

Unlock the best of we

Access a world of Web3 opportunities securely with a single sign-in.

We need atleast 2 **Unique Humanity Score** to proceed mining:



Let's increase that score

Here's some tips on how to raise your score to a minimum of 20.

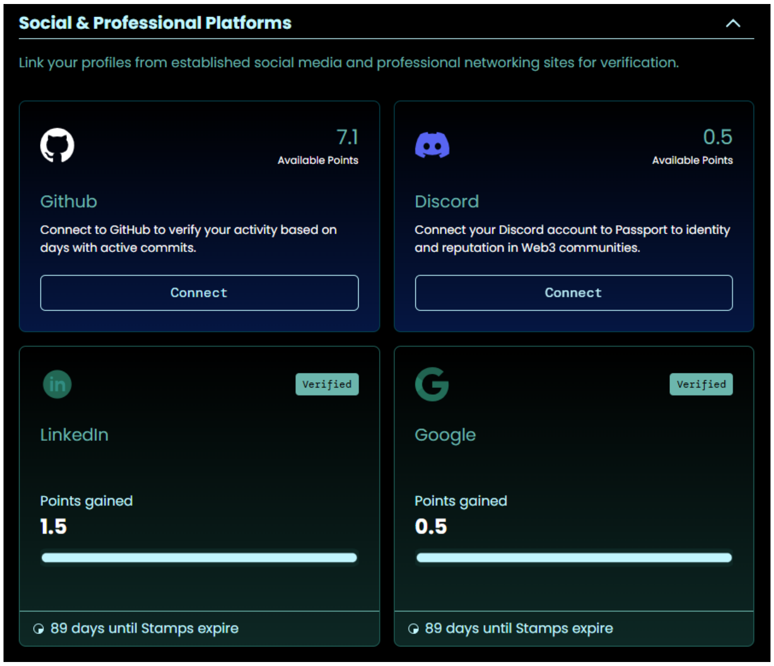
Add Stamps

Blockchain & Crypto Networks

Connect your blockchain-based profiles and assets to prove your identity.

Scroll down to Social & Professional Platforms, and link your Linkedin & Google account:

[These sites are safe, so feel free to go ahead]



Social & Professional Platforms

Link your profiles from established social media and professional networking sites for verification.

Github — 7.1 Available Points

Connect to GitHub to verify your activity based on days with active commits.

Discord — 0.5 Available Points

Connect your Discord account to Passport to identity and reputation in Web3 communities.

LinkedIn — Verified

Points gained
1.5

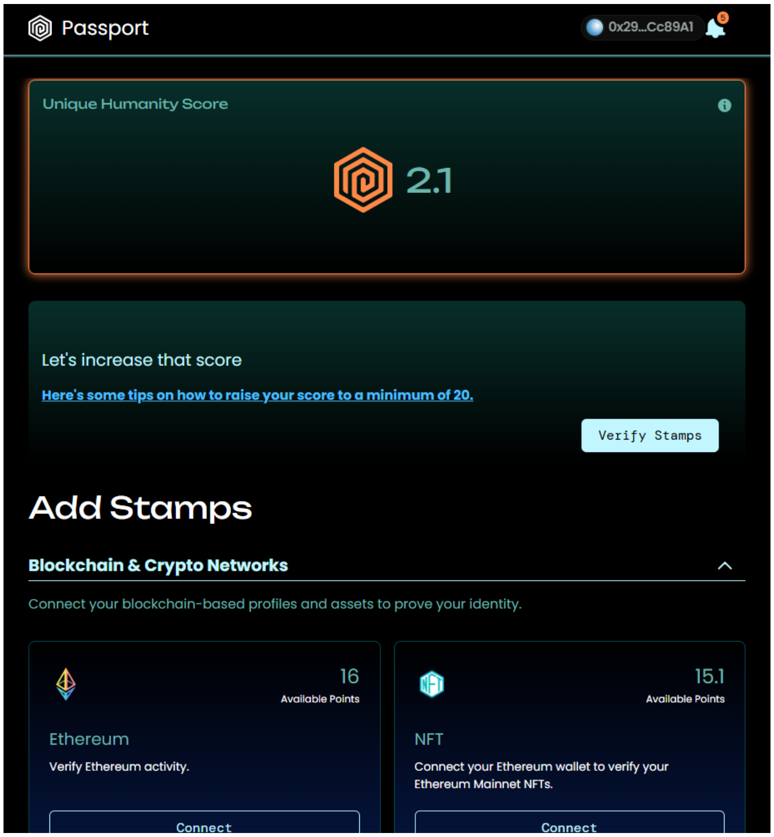
89 days until Stamps expire

Google — Verified

Points gained
0.5

89 days until Stamps expire

So now we have **Unique Humanity Score** of **2.1**, we are good to go for mining:



Unique Humanity Score

2.1

Let's increase that score

Here's some tips on how to raise your score to a minimum of 20.

Add Stamps

Blockchain & Crypto Networks

Connect your blockchain-based profiles and assets to prove your identity.

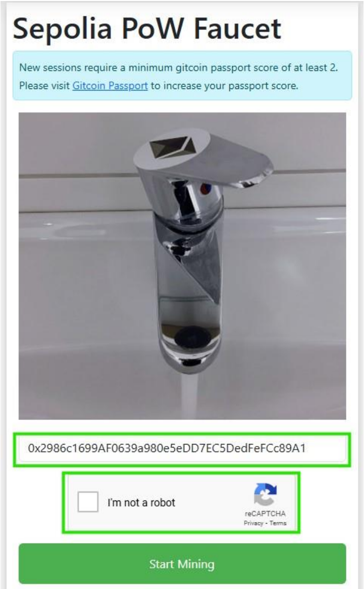
Ethereum — 16 Available Points

Verify Ethereum activity.

NFT — 15.1 Available Points

Connect your Ethereum wallet to verify your Ethereum Mainnet NFTs.

Now go to https://sepolia-faucet.pk910.de/#/ (Paste the wallet address, complete the captcha, and click **Start Mining**):

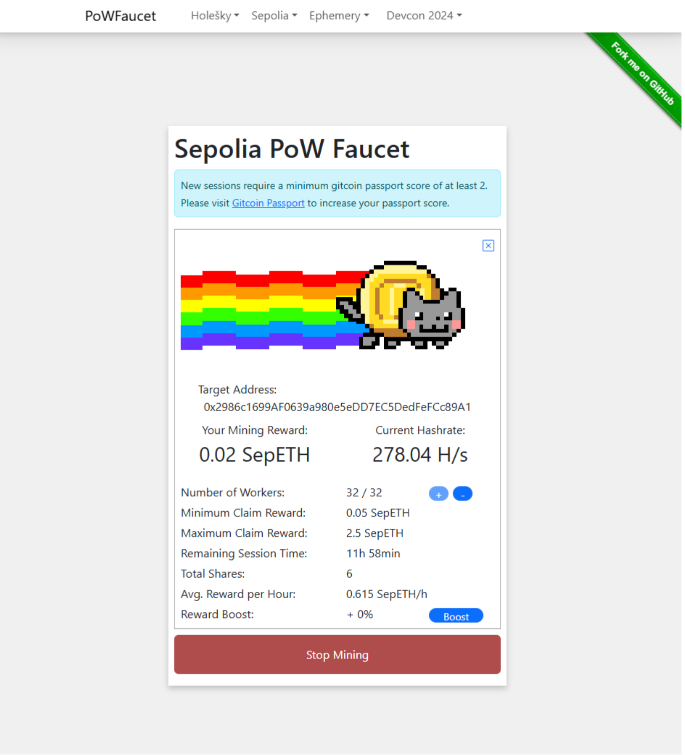


Sepolia PoW Faucet

New sessions require a minimum gitcoin passport score of at least 2. Please visit Gitcoin Passport to increase your passport score.

0x2986c1699AF0639a980e5eDD7EC5DedFeFCc89A1

It should now start mining. The mining rate depends on the Hashrate which relies on the PC's configuration. It's recommended to close all other heavy softwares before mining for better efficiency. Minimum 0.05 ETH needs to be mine for withdrawal, and maximum 2.5 ETH can be mined in one session. New sessions can be started as many times as we want. So no worries.



Sepolia PoW Faucet

New sessions require a minimum gitcoin passport score of at least 2. Please visit Gitcoin Passport to increase your passport score.

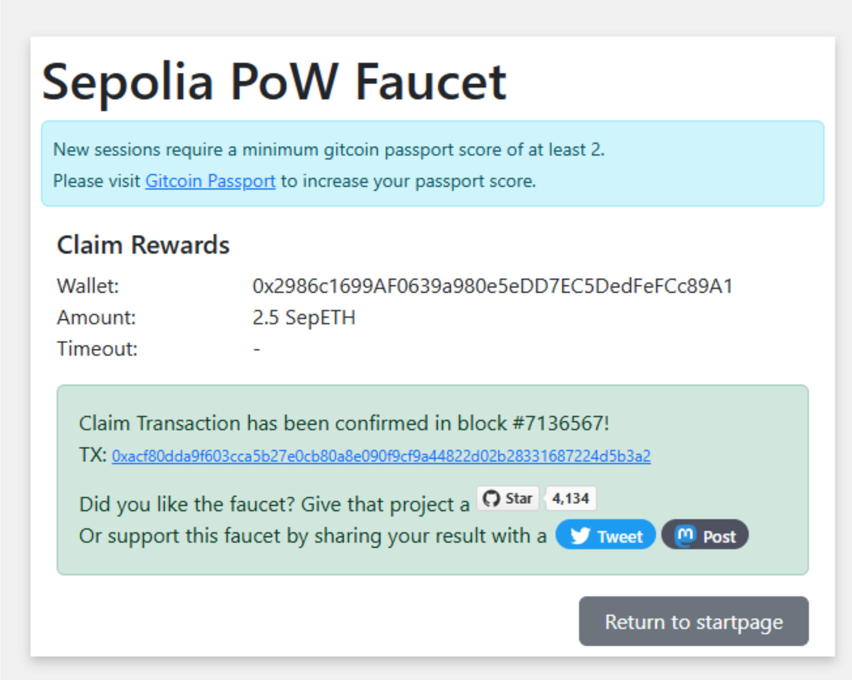
Target Address:
0x2986c1699AF0639a980e5eDD7EC5DedFeFCc89A1

Your Mining Reward: 0.02 SepETH

Current Hashrate: 278.04 H/s

Number of Workers:	32 / 32
Minimum Claim Reward:	0.05 SepETH
Maximum Claim Reward:	2.5 SepETH
Remaining Session Time:	11h 58min
Total Shares:	6
Avg. Reward per Hour:	0.615 SepETH/h
Reward Boost:	+ 0%

After 2.5 ETH has been mined the mining process will be automatically terminated, then click on claim reward, and we'll see the claim transaction has been confirmed:



Sepolia PoW Faucet

New sessions require a minimum gitcoin passport score of at least 2.
Please visit Gitcoin Passport to increase your passport score.

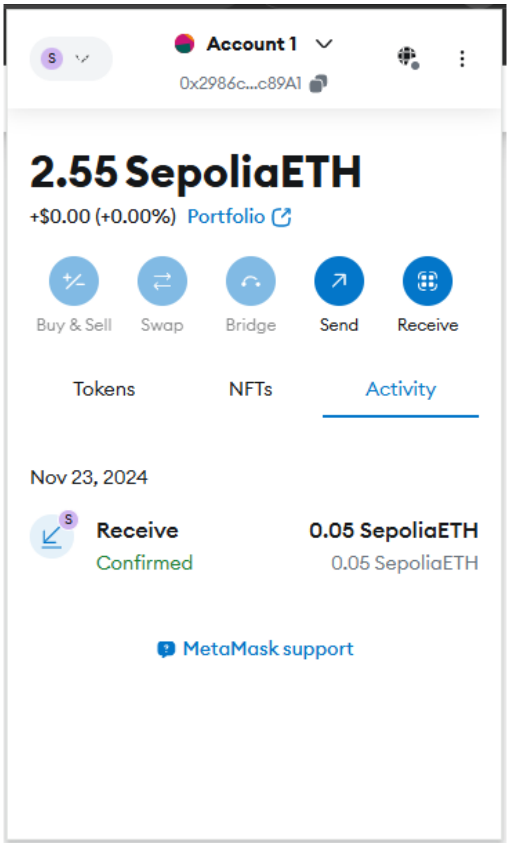
Claim Rewards

Wallet: 0x2986c1699AF0639a980e5eDD7EC5DedFeFCc89A1
Amount: 2.5 SepETH
Timeout: -

Claim Transaction has been confirmed in block #7136567!
TX: 0xacf80dda9f603cca5b27e0cb80a8e090f9cf9a44822d02b28331687224d5b3a2

Did you like the faucet? Give that project a Star 4,134
Or support this faucet by sharing your result with a Tweet Post

Let's check our wallet:

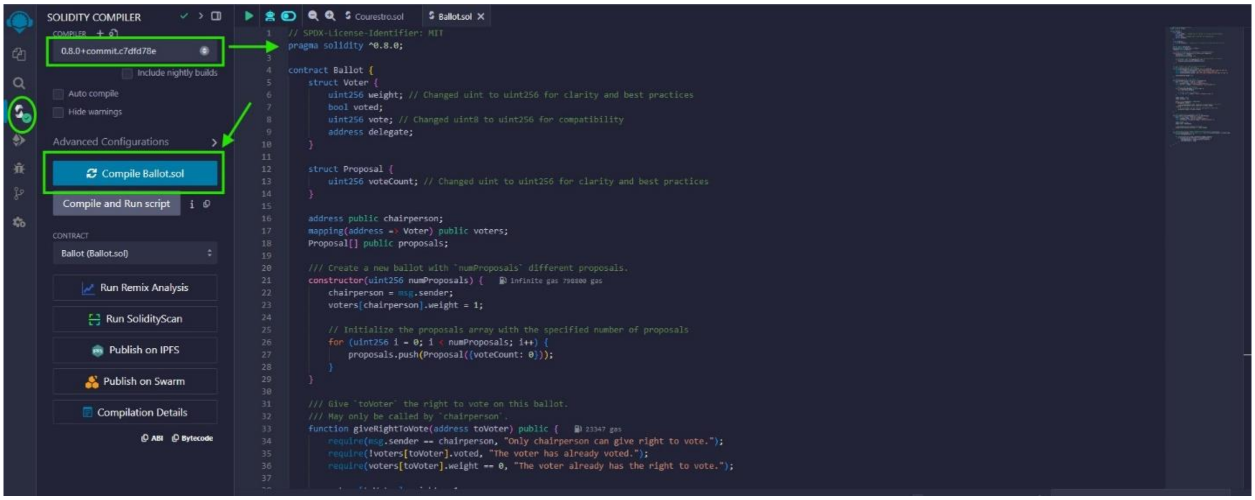


2.55 SepoliaETH
+$0.00 (+0.00%) Portfolio

Nov 23, 2024

Receive 0.05 SepoliaETH
Confirmed 0.05 SepoliaETH

Let’s compile our ballot.sol now in solidity version 0.8.0:



COMPILER: 0.8.0+commit.c7dfd78e

□ Include nightly builds

□ Auto compile

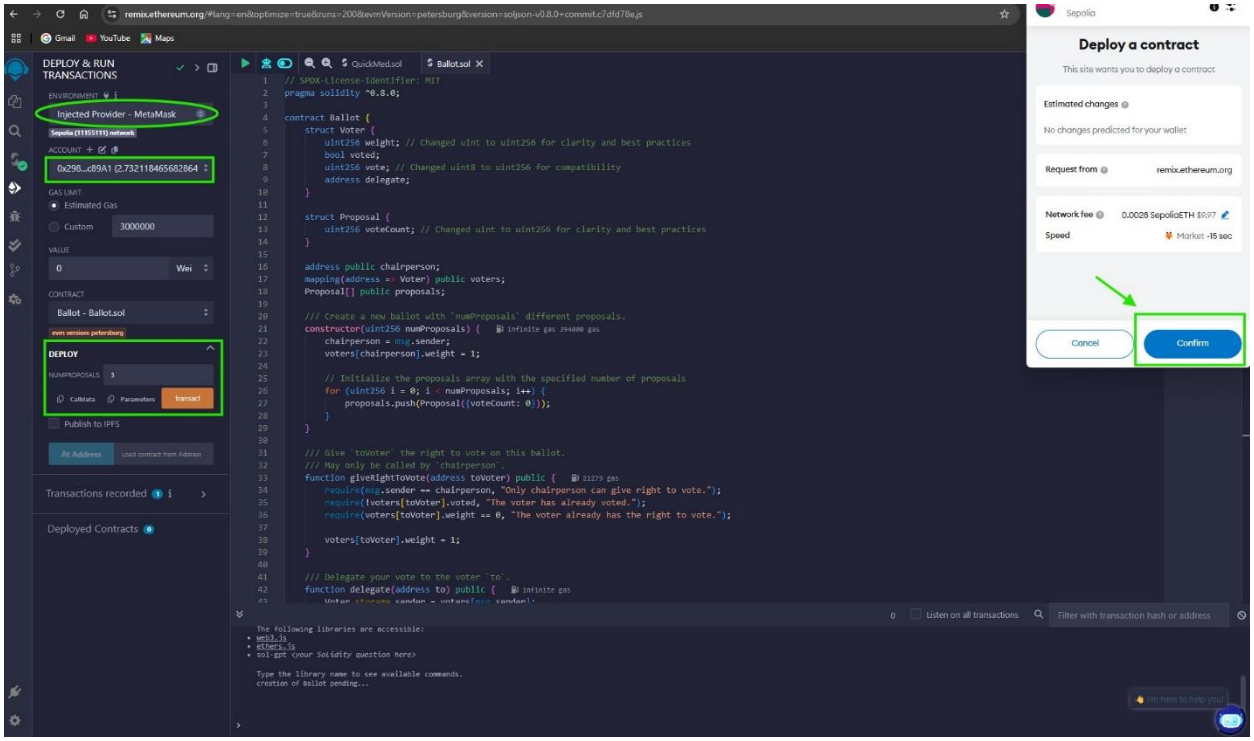
□ Hide warnings

Advanced Configurations

CONTRACT: Ballot (Ballot.sol)

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract Ballot {
    struct Voter {
        uint256 weight; // Changed uint to uint256 for clarity and best practices
        bool voted;
        uint256 vote; // Changed uint8 to uint256 for compatibility
        address delegate;
    }

    struct Proposal {
        uint256 voteCount; // Changed uint to uint256 for clarity and best practices
    }

    address public chairperson;
    mapping(address => Voter) public voters;
    Proposal[] public proposals;

    /// Create a new ballot with `numProposals` different proposals.
    constructor(uint256 numProposals) {    infinite gas 798800 gas
        chairperson = msg.sender;
        voters[chairperson].weight = 1;

        // Initialize the proposals array with the specified number of proposals
        for (uint256 i = 0; i < numProposals; i++) {
            proposals.push(Proposal({voteCount: 0}));
        }
    }

    /// Give `toVoter` the right to vote on this ballot.
    /// May only be called by `chairperson`.
    function giveRightToVote(address toVoter) public {    23347 gas
        require(msg.sender == chairperson, "Only chairperson can give right to vote.");
        require(!voters[toVoter].voted, "The voter has already voted.");
        require(voters[toVoter].weight == 0, "The voter already has the right to vote.");
```

Let’s deploy the smart contract on **Sepolia Testnet** on **Injected Provider -MetaMask** environment:



remix.ethereum.org/#lang=en&optimize=true&runs=200&evmVersion=petersburg&version=soljson-v0.8.0+commit.c7dfd78e.js

ENVIRONMENT: Injected Provider - MetaMask

Sepolia (11155111) network

ACCOUNT: 0x298...c89A1 (2.732118465682864

GAS LIMIT: ● Estimated Gas ○ Custom 3000000

VALUE: 0 Wei

CONTRACT: Ballot - Ballot.sol

evm versions petersburg

DEPLOY

NUMPROPOSALS: 3

□ Publish to IPFS

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract Ballot {
    struct Voter {
        uint256 weight; // Changed uint to uint256 for clarity and best practices
        bool voted;
        uint256 vote; // Changed uint8 to uint256 for compatibility
        address delegate;
    }

    struct Proposal {
        uint256 voteCount; // Changed uint to uint256 for clarity and best practices
    }

    address public chairperson;
    mapping(address => Voter) public voters;
    Proposal[] public proposals;

    /// Create a new ballot with `numProposals` different proposals.
    constructor(uint256 numProposals) {    infinite gas 394000 gas
        chairperson = msg.sender;
        voters[chairperson].weight = 1;

        // Initialize the proposals array with the specified number of proposals
        for (uint256 i = 0; i < numProposals; i++) {
            proposals.push(Proposal({voteCount: 0}));
        }
    }

    /// Give `toVoter` the right to vote on this ballot.
    /// May only be called by `chairperson`.
    function giveRightToVote(address toVoter) public {    21279 gas
        require(msg.sender == chairperson, "Only chairperson can give right to vote.");
        require(!voters[toVoter].voted, "The voter has already voted.");
        require(voters[toVoter].weight == 0, "The voter already has the right to vote.");

        voters[toVoter].weight = 1;
    }

    /// Delegate your vote to the voter `to`.
    function delegate(address to) public {    infinite gas
        Voter storage sender = voters[msg.sender];
```

```
The following libraries are accessible:
• web3.js
• ethers.js
• sol-gpt <your Solidity question here>

Type the library name to see available commands.
creation of Ballot pending...
```

Sepolia

Deploy a contract

This site wants you to deploy a contract

Estimated changes

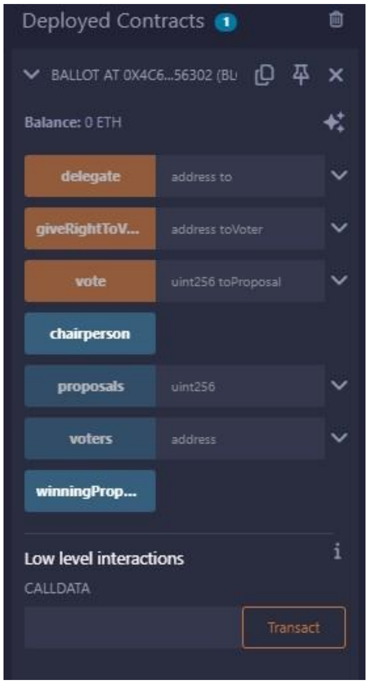
No changes predicted for your wallet

Request from: remix.ethereum.org

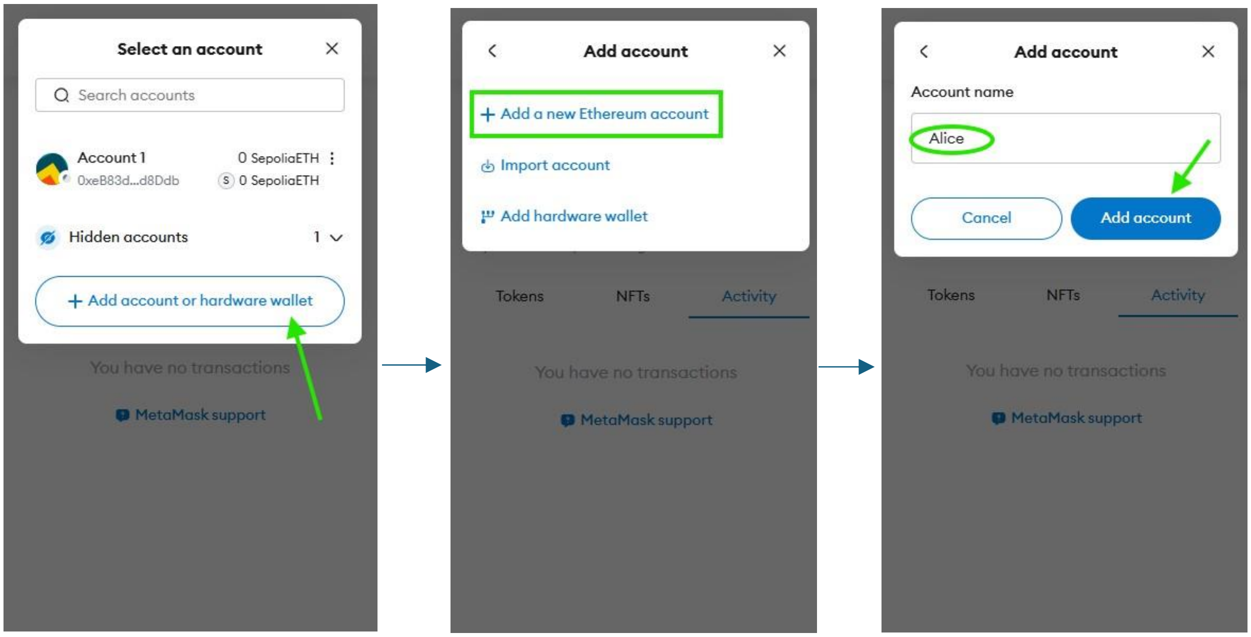
Network fee: 0.0028 SepoliaETH $9.97

Speed: Market -15 sec

After deploying, the contracts functional output should look like this:

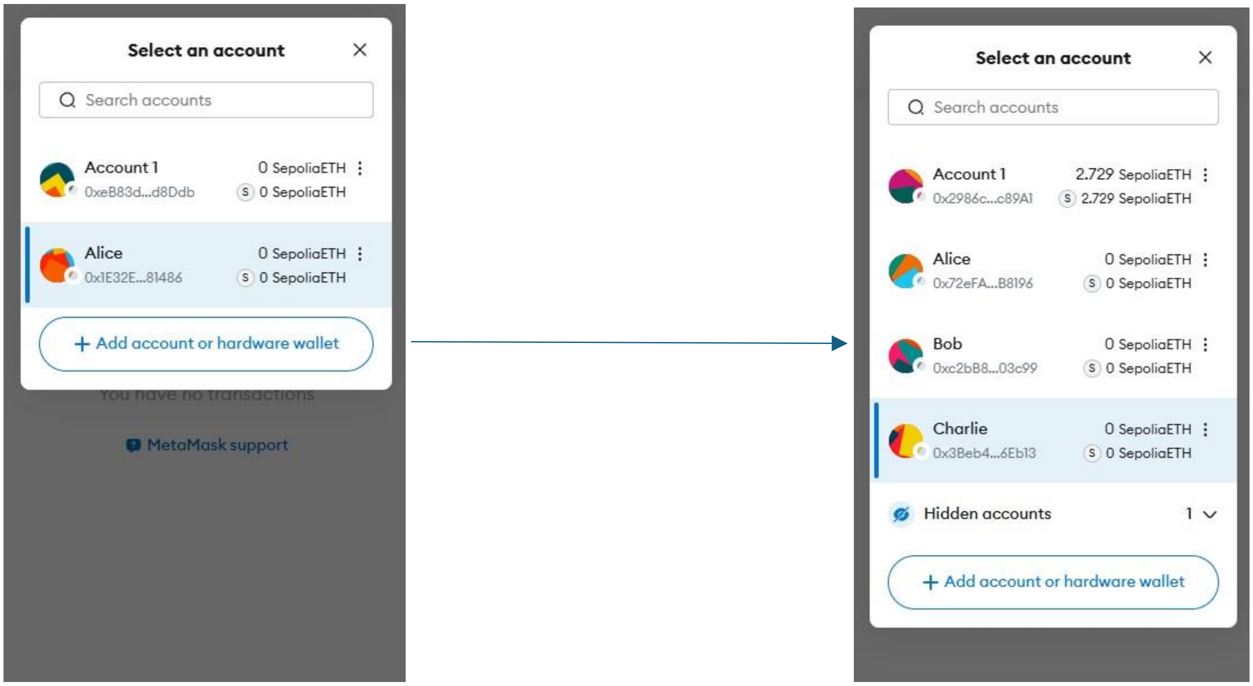


Now, let’s create some accounts, since the contract involves multiple users to interact:



Account name: Alice

Just like **Alice**, let’s create wallets for **Bob**, and **Charlie**. Considering Account 1 as the **Chairperson**, since it was used to deploy this smart contract:



Select an account

Search accounts

Account 1 0xeB83d...d8Ddb	0 SepoliaETH S 0 SepoliaETH	⋮
Alice 0x1E32E...81486	0 SepoliaETH S 0 SepoliaETH	⋮

+ Add account or hardware wallet

You have no transactions

MetaMask support

Select an account

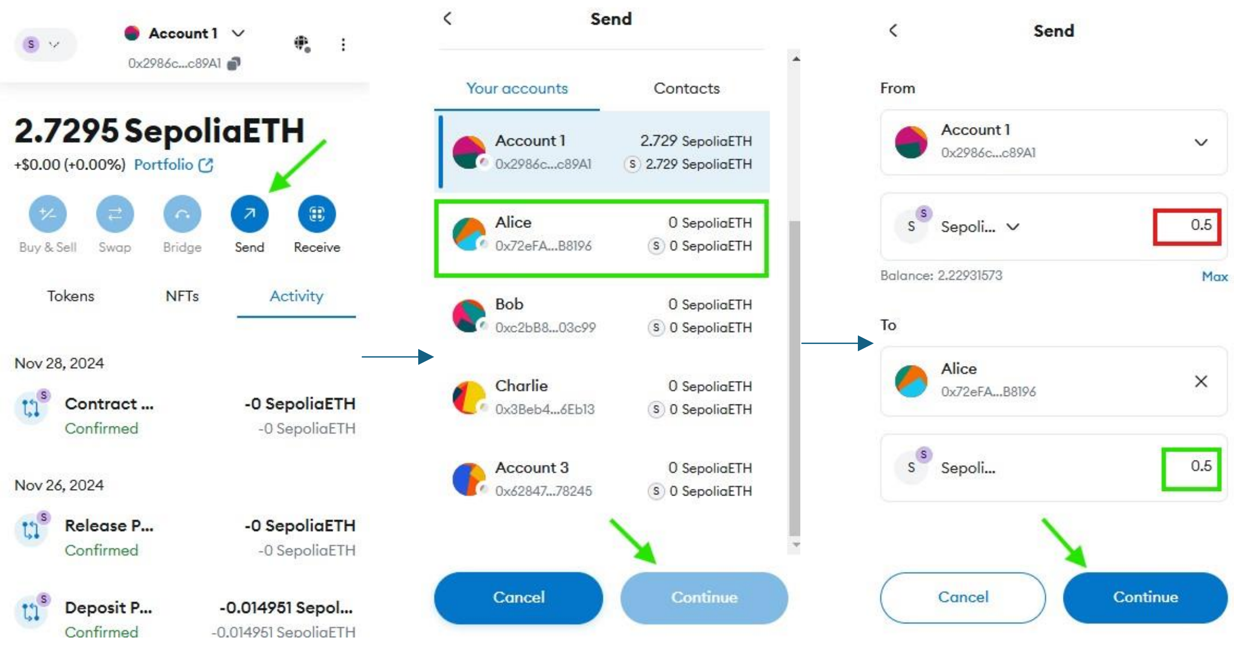
Search accounts

Account 1 0x2986c...c89A1	2.729 SepoliaETH S 2.729 SepoliaETH	⋮
Alice 0x72eFA...B8196	0 SepoliaETH S 0 SepoliaETH	⋮
Bob 0xc2bB8...03c99	0 SepoliaETH S 0 SepoliaETH	⋮
Charlie 0x3Beb4...6Eb13	0 SepoliaETH S 0 SepoliaETH	⋮

Hidden accounts 1

+ Add account or hardware wallet

Now, we’d also need some ETH on other newly created wallets as well. Let’s transfer some ETH. Since I had plenty of ETH I transferred 0.5 ETH to each of these new wallets. In case you didn’t mine some more ETH, just transfer 0.01 ETH to those three newly created wallets. 0.01 ETH on each wallet is also enough for this task.



Account 1

0x2986c...c89A1

2.7295 SepoliaETH

+$0.00 (+0.00%) Portfolio

Nov 28, 2024

Contract ... Confirmed	-0 SepoliaETH -0 SepoliaETH

Nov 26, 2024

Release P... Confirmed	-0 SepoliaETH -0 SepoliaETH
Deposit P... Confirmed	-0.014951 Sepol... -0.014951 SepoliaETH

Send

Your accounts Contacts

Account 1 0x2986c...c89A1	2.729 SepoliaETH S 2.729 SepoliaETH
Alice 0x72eFA...B8196	0 SepoliaETH S 0 SepoliaETH
Bob 0xc2bB8...03c99	0 SepoliaETH S 0 SepoliaETH
Charlie 0x3Beb4...6Eb13	0 SepoliaETH S 0 SepoliaETH
Account 3 0x62847...78245	0 SepoliaETH S 0 SepoliaETH

Cancel Continue

Send

From

Account 1
0x2986c...c89A1

Sepoli... 0.5

Balance: 2.22931573 Max

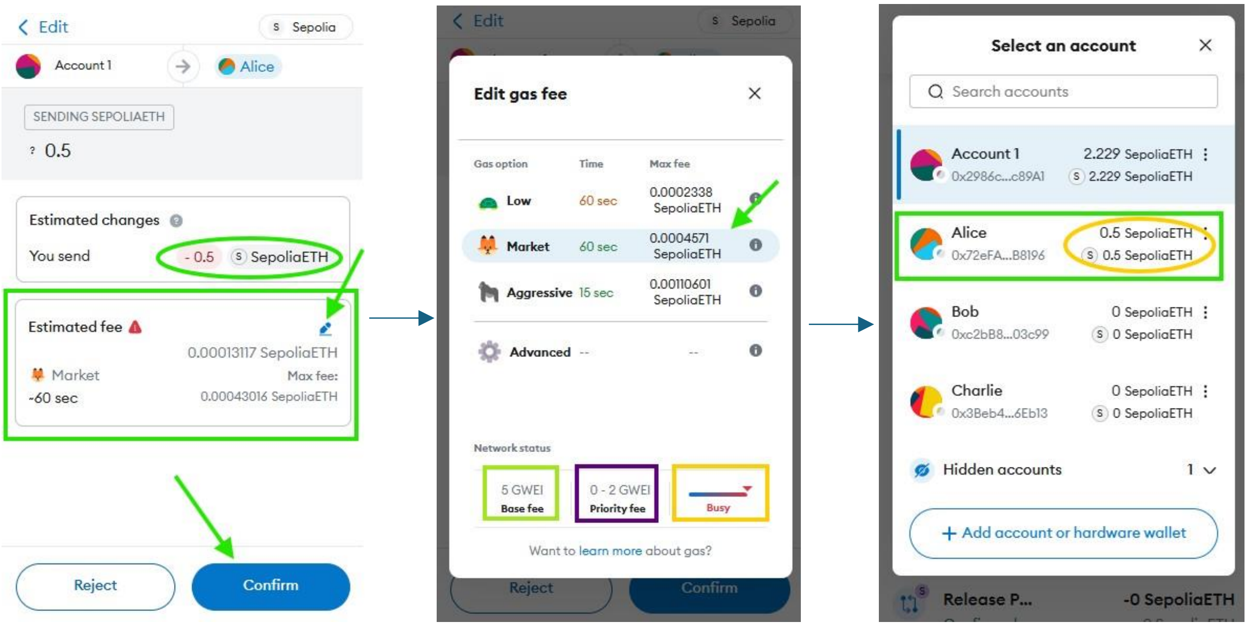
To

Alice
0x72eFA...B8196

Sepoli... 0.5

Cancel Continue

During the transfer you can also configure the transaction mining time. In case of aggressive mining we'll have to pay 0-2 GWEI extra, where the base fee is 5 GWEI:



Account 1 → Alice

SENDING SEPOLIAETH

0.5

Estimated changes

You send - 0.5 SepoliaETH

Estimated fee 0.00013117 SepoliaETH

Market Max fee:

~60 sec 0.00043016 SepoliaETH

Edit gas fee

Gas option	Time	Max fee
Low	60 sec	0.0002338 SepoliaETH
Market	60 sec	0.0004571 SepoliaETH
Aggressive	15 sec	0.00110601 SepoliaETH
Advanced	--	--

Network status

5 GWEI	0 - 2 GWEI	
Base fee	Priority fee	Busy

Want to learn more about gas?

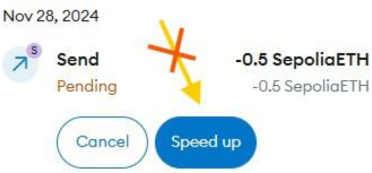
Select an account

Search accounts

Account 1	0x2986c...c89A1	2.229 SepoliaETH
Alice	0x72eFA...B8196	0.5 SepoliaETH
Bob	0xc2bB8...03c99	0 SepoliaETH
Charlie	0x3Beb4...6Eb13	0 SepoliaETH

Hidden accounts 1

Do not press **Speed Up** while the transaction is pending unless we want to spend some extra ETH.

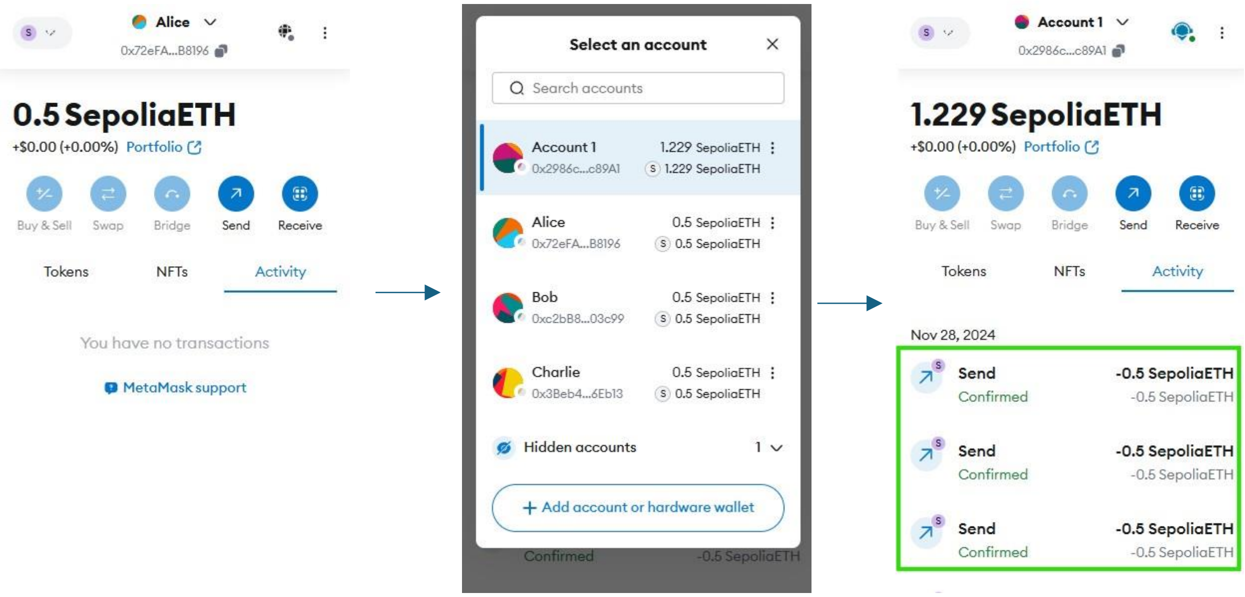


Nov 28, 2024

Send -0.5 SepoliaETH

Pending -0.5 SepoliaETH

All of the accounts have ETH on their wallet now:



Alice 0x72eFA...B8196

0.5 SepoliaETH

+$0.00 (+0.00%) Portfolio

You have no transactions

Select an account

Search accounts

Account 1	0x2986c...c89A1	1.229 SepoliaETH
Alice	0x72eFA...B8196	0.5 SepoliaETH
Bob	0xc2bB8...03c99	0.5 SepoliaETH
Charlie	0x3Beb4...6Eb13	0.5 SepoliaETH

Hidden accounts 1

Account 1 0x2986c...c89A1

1.229 SepoliaETH

+$0.00 (+0.00%) Portfolio

Nov 28, 2024

Send -0.5 SepoliaETH

Confirmed -0.5 SepoliaETH

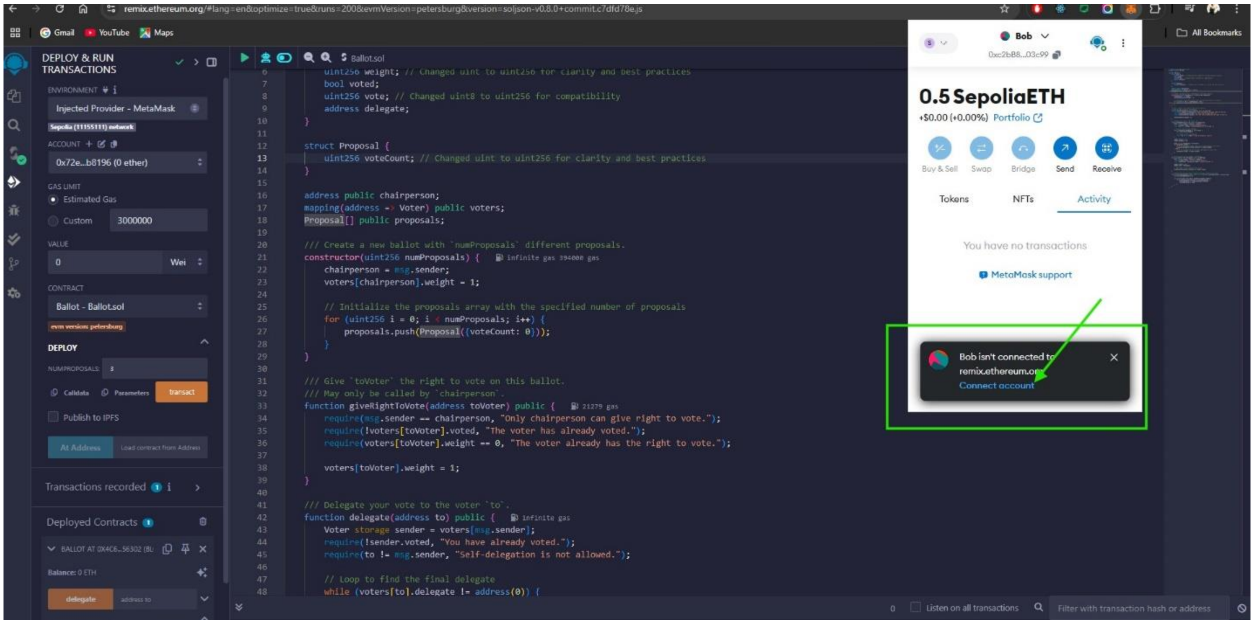
Send -0.5 SepoliaETH

Confirmed -0.5 SepoliaETH

Send -0.5 SepoliaETH

Confirmed -0.5 SepoliaETH

Connect all the new accounts to the Remix IDE:



remix.ethereum.org/#lang=en&optimize=true&runs=200&evmVersion=petersburg&version=soljson-v0.8.0+commit.c7dfd78e.js

DEPLOY & RUN TRANSACTIONS

ENVIRONMENT: Injected Provider - MetaMask

Sepolia (11155111) network

ACCOUNT: 0x72e...b8196 (0 ether)

GAS LIMIT: ● Estimated Gas ○ Custom 3000000

VALUE: 0 Wei

CONTRACT: Ballot - Ballot.sol

evm version: petersburg

DEPLOY

NUMPROPOSALS: 3

□ Publish to IPFS

Transactions recorded 1

Deployed Contracts 1

BALLOT AT 0X4C6...56302 (BL

Balance: 0 ETH

Ballot.sol

```
        uint256 weight; // Changed uint to uint256 for clarity and best practices
        bool voted;
        uint256 vote; // Changed uint8 to uint256 for compatibility
        address delegate;
    }

    struct Proposal {
        uint256 voteCount; // Changed uint to uint256 for clarity and best practices
    }

    address public chairperson;
    mapping(address => Voter) public voters;
    Proposal[] public proposals;

    /// Create a new ballot with `numProposals` different proposals.
    constructor(uint256 numProposals) {    infinite gas 394000 gas
        chairperson = msg.sender;
        voters[chairperson].weight = 1;

        // Initialize the proposals array with the specified number of proposals
        for (uint256 i = 0; i < numProposals; i++) {
            proposals.push(Proposal({voteCount: 0}));
        }
    }

    /// Give `toVoter` the right to vote on this ballot.
    /// May only be called by `chairperson`.
    function giveRightToVote(address toVoter) public {    21279 gas
        require(msg.sender == chairperson, "Only chairperson can give right to vote.");
        require(!voters[toVoter].voted, "The voter has already voted.");
        require(voters[toVoter].weight == 0, "The voter already has the right to vote.");

        voters[toVoter].weight = 1;
    }

    /// Delegate your vote to the voter `to`.
    function delegate(address to) public {    infinite gas
        Voter storage sender = voters[msg.sender];
        require(!sender.voted, "You have already voted.");
        require(to != msg.sender, "Self-delegation is not allowed.");

        // Loop to find the final delegate
        while (voters[to].delegate != address(0)) {
```

Bob

0xc2bB8...03c99

0.5 SepoliaETH

+$0.00 (+0.00%) Portfolio

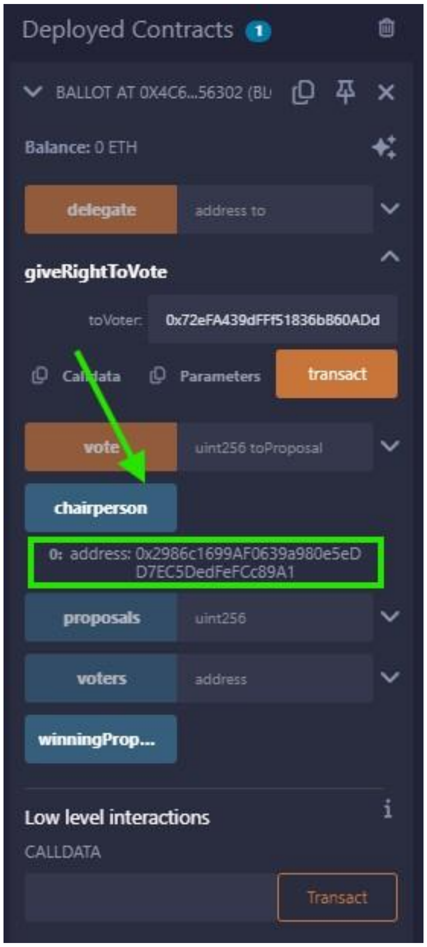
You have no transactions

MetaMask support

Bob isn't connected to remix.ethereum.org

Connect account

Click on Chairperson, and we'll see Account 1's wallet address is the Chairperson:



Deployed Contracts 1

BALLOT AT 0X4C6...56302 (BL

Balance: 0 ETH

giveRightToVote

toVoter: 0x72eFA439dFF51836bB60ADd

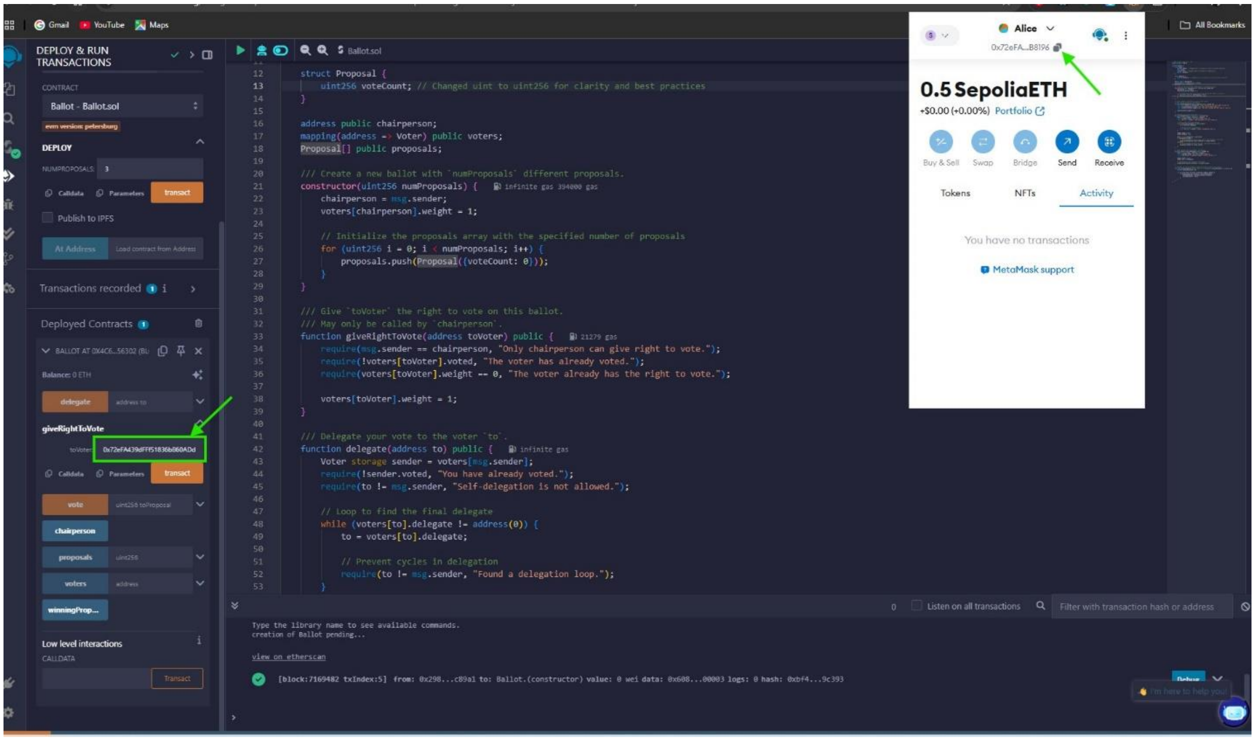
chairperson

0: address: 0x2986c1699AF0639a980e5eDD7EC5DedFeFCc89A1

Low level interactions

CALLDATA

Now copy Alice’s wallet address and give her the right to vote, but before clicking **Transact** make sure to change the active wallet in MetaMask from **Alice** to **Account 1**:



NUMPROPOSALS: 3

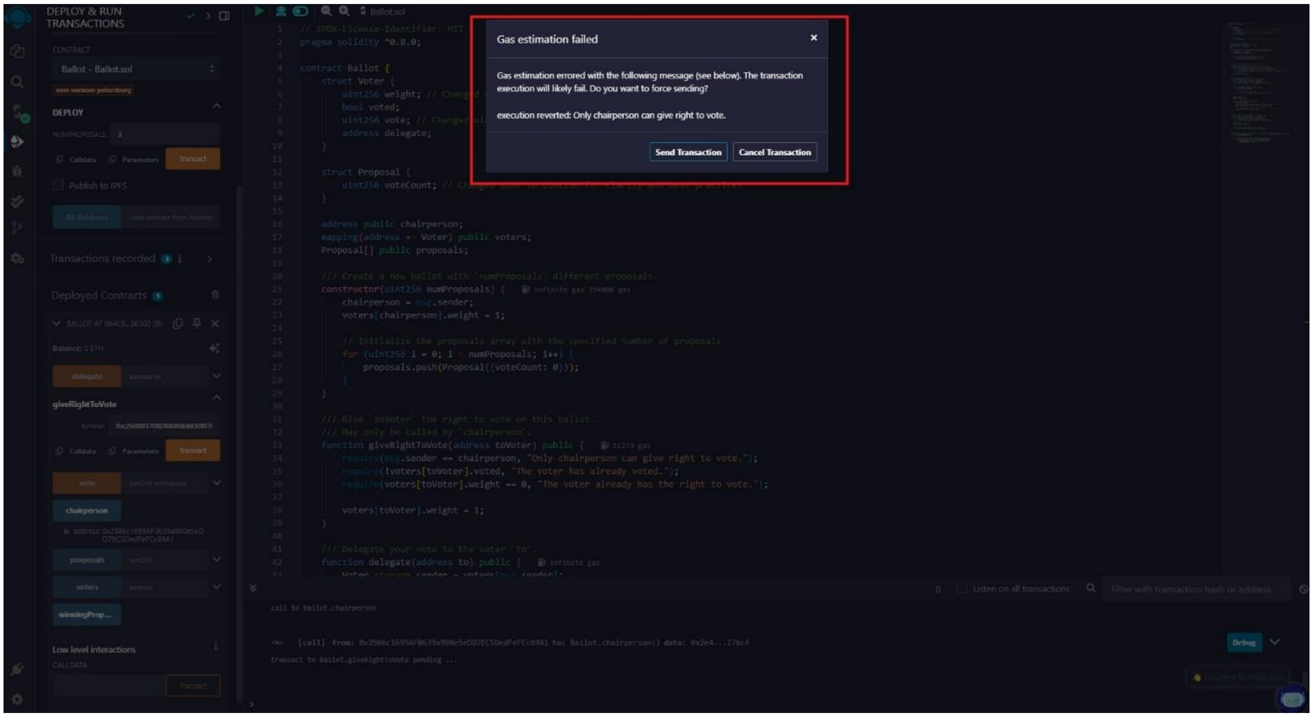
toVoter: 0x72eFA439dFFf51836b860ADd

0.5 SepoliaETH
+$0.00 (+0.00%) Portfolio

```
    struct Proposal {
        uint256 voteCount; // Changed uint to uint256 for clarity and best practices
    }

    address public chairperson;
    mapping(address => Voter) public voters;
    Proposal[] public proposals;

    /// Create a new ballot with `numProposals` different proposals.
    constructor(uint256 numProposals) {    infinite gas 394000 gas
        chairperson = msg.sender;
        voters[chairperson].weight = 1;

        // Initialize the proposals array with the specified number of proposals
        for (uint256 i = 0; i < numProposals; i++) {
            proposals.push(Proposal({voteCount: 0}));
        }
    }

    /// Give `toVoter` the right to vote on this ballot.
    /// May only be called by `chairperson`.
    function giveRightToVote(address toVoter) public {    21279 gas
        require(msg.sender == chairperson, "Only chairperson can give right to vote.");
        require(!voters[toVoter].voted, "The voter has already voted.");
        require(voters[toVoter].weight == 0, "The voter already has the right to vote.");

        voters[toVoter].weight = 1;
    }

    /// Delegate your vote to the voter `to`.
    function delegate(address to) public {    infinite gas
        Voter storage sender = voters[msg.sender];
        require(!sender.voted, "You have already voted.");
        require(to != msg.sender, "Self-delegation is not allowed.");

        // Loop to find the final delegate
        while (voters[to].delegate != address(0)) {
            to = voters[to].delegate;

            // Prevent cycles in delegation
            require(to != msg.sender, "Found a delegation loop.");
        }
```

```
Type the library name to see available commands.
creation of Ballot pending...

view on etherscan

[block:7169482 txIndex:5] from: 0x298...c89a1 to: Ballot.(constructor) value: 0 wei data: 0x608...00003 logs: 0 hash: 0xbf4...9c393
```

In case, forgot to switch back to Account; we’ll see an error since only the Chairperson can give the right to vote. Do it correctly for Bob, and Charlie as well:



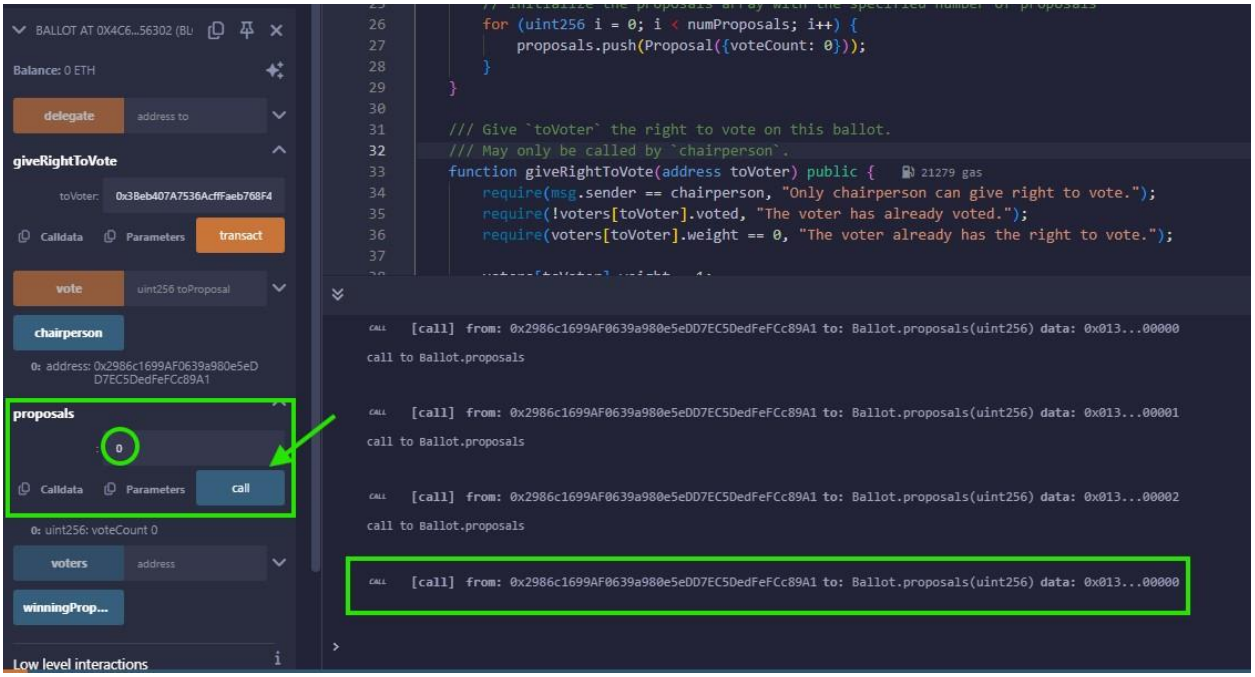
Gas estimation failed

Gas estimation errored with the following message (see below). The transaction execution will likely fail. Do you want to force sending?

execution reverted: Only chairperson can give right to vote.

```
call to Ballot.chairperson

[call] from: 0x2986c1699AF0639a980e5eDD7EC5DedFeFCc89A1 to: Ballot.chairperson() data: 0x2e4...176cf
transact to Ballot.giveRightToVote pending ...
```

If we call the Proposal function with the index number of the proposal, we will see the number of votes it received:



Balance: 0 ETH

toVoter: 0x3Beb407A7536AcffFaeb768F4

0: address: 0x2986c1699AF0639a980e5eD D7EC5DedFeFCc89A1

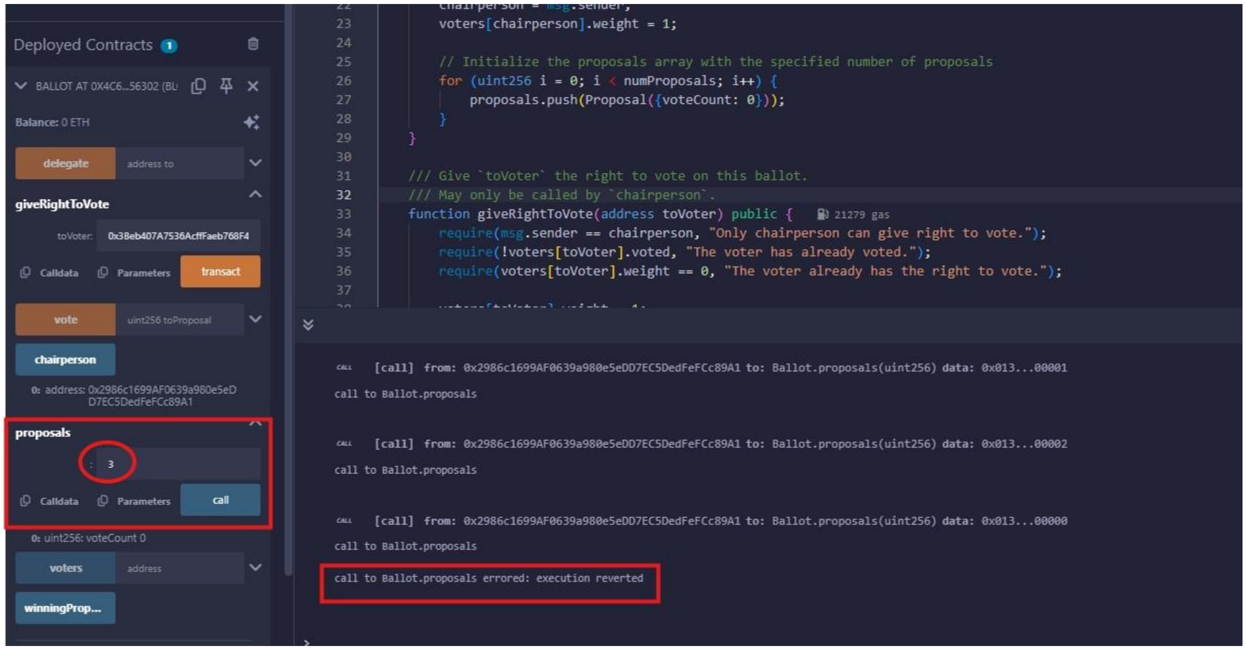
proposals: 0

0: uint256: voteCount 0

```
        for (uint256 i = 0; i < numProposals; i++) {
            proposals.push(Proposal({voteCount: 0}));
        }
    }

    /// Give `toVoter` the right to vote on this ballot.
    /// May only be called by `chairperson`.
    function giveRightToVote(address toVoter) public {     21279 gas
        require(msg.sender == chairperson, "Only chairperson can give right to vote.");
        require(!voters[toVoter].voted, "The voter has already voted.");
        require(voters[toVoter].weight == 0, "The voter already has the right to vote.");
```

```
[call] from: 0x2986c1699AF0639a980e5eDD7EC5DedFeFCc89A1 to: Ballot.proposals(uint256) data: 0x013...00000
call to Ballot.proposals

[call] from: 0x2986c1699AF0639a980e5eDD7EC5DedFeFCc89A1 to: Ballot.proposals(uint256) data: 0x013...00001
call to Ballot.proposals

[call] from: 0x2986c1699AF0639a980e5eDD7EC5DedFeFCc89A1 to: Ballot.proposals(uint256) data: 0x013...00002
call to Ballot.proposals

[call] from: 0x2986c1699AF0639a980e5eDD7EC5DedFeFCc89A1 to: Ballot.proposals(uint256) data: 0x013...00000
```

The proposals are stored in an array. So, if the total number of proposals is 3, then writing 2 here indicates proposal number 3. And writing 3 would throw an error, since it would exceed the array's max index, which is 2 (0 represents proposal 1).



Balance: 0 ETH

toVoter: 0x3Beb407A7536AcffFaeb768F4

0: address: 0x2986c1699AF0639a980e5eD D7EC5DedFeFCc89A1

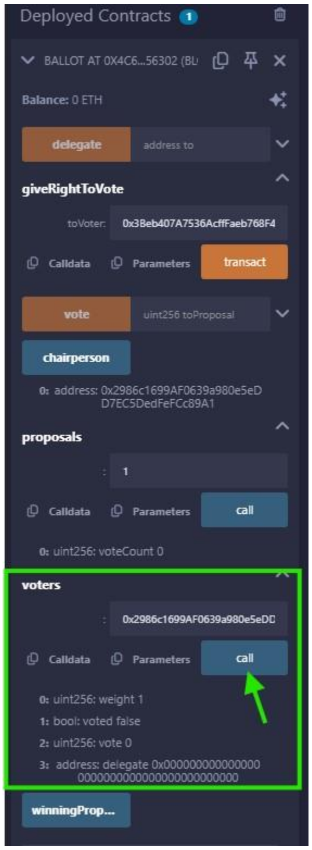
proposals: 3

0: uint256: voteCount 0

```
        voters[chairperson].weight = 1;

        // Initialize the proposals array with the specified number of proposals
        for (uint256 i = 0; i < numProposals; i++) {
            proposals.push(Proposal({voteCount: 0}));
        }
    }

    /// Give `toVoter` the right to vote on this ballot.
    /// May only be called by `chairperson`.
    function giveRightToVote(address toVoter) public {     21279 gas
        require(msg.sender == chairperson, "Only chairperson can give right to vote.");
        require(!voters[toVoter].voted, "The voter has already voted.");
        require(voters[toVoter].weight == 0, "The voter already has the right to vote.");
```

```
[call] from: 0x2986c1699AF0639a980e5eDD7EC5DedFeFCc89A1 to: Ballot.proposals(uint256) data: 0x013...00001
call to Ballot.proposals

[call] from: 0x2986c1699AF0639a980e5eDD7EC5DedFeFCc89A1 to: Ballot.proposals(uint256) data: 0x013...00002
call to Ballot.proposals

[call] from: 0x2986c1699AF0639a980e5eDD7EC5DedFeFCc89A1 to: Ballot.proposals(uint256) data: 0x013...00000
call to Ballot.proposals
call to Ballot.proposals errored: execution reverted
```

Paste **Alice's address** on **voters** function and **call** it to see her status as a voter:



Deployed Contracts 1

BALLOT AT 0X4C6...56302 (BL

Balance: 0 ETH

delegate: address to

giveRightToVote

toVoter: 0x3Beb407A7536AcffFaeb768F4

vote: uint256 toProposal

chairperson

0: address: 0x2986c1699AF0639a980e5eD D7EC5DedFeFCc89A1

proposals

: 1

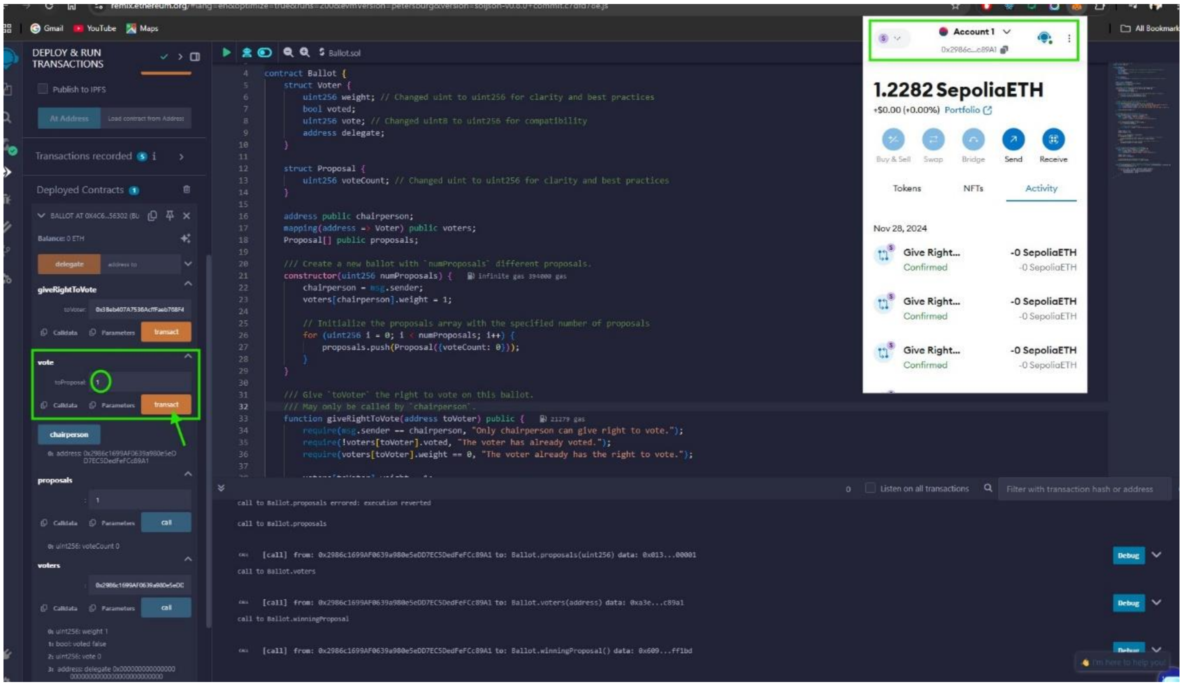
0: uint256: voteCount 0

voters

: 0x2986c1699AF0639a980e5eDD

0: uint256: weight 1
1: bool: voted false
2: uint256: vote 0
3: address: delegate 0x000000000000000 0000000000000000000000000

Now, switching back to **Account 1 (Chairperson)**, and voting for proposal 2, thus writing 1, and clicking **Transact**:



DEPLOY & RUN TRANSACTIONS

☐ Publish to IPFS

Transactions recorded 5

Deployed Contracts 1

BALLOT AT 0X4C6...56302 (BL

Balance: 0 ETH

giveRightToVote

toVoter: 0x3Beb407A7536AcffFaeb768F4

vote

toProposal: 1

chairperson

0: address: 0x2986c1699AF0639a980e5eD D7EC5DedFeFCc89A1

proposals

: 1

0: uint256: voteCount 0

voters

: 0x2986c1699AF0639a980e5eDD

0: uint256: weight 1
1: bool: voted false
2: uint256: vote 0
3: address: delegate 0x000000000000000 0000000000000000000000000

Ballot.sol

```
contract Ballot {
    struct Voter {
        uint256 weight; // Changed uint to uint256 for clarity and best practices
        bool voted;
        uint256 vote; // Changed uint8 to uint256 for compatibility
        address delegate;
    }

    struct Proposal {
        uint256 voteCount; // Changed uint to uint256 for clarity and best practices
    }

    address public chairperson;
    mapping(address => Voter) public voters;
    Proposal[] public proposals;

    /// Create a new ballot with `numProposals` different proposals.
    constructor(uint256 numProposals) {    infinite gas 394000 gas
        chairperson = msg.sender;
        voters[chairperson].weight = 1;

        // Initialize the proposals array with the specified number of proposals
        for (uint256 i = 0; i < numProposals; i++) {
            proposals.push(Proposal({voteCount: 0}));
        }
    }

    /// Give `toVoter` the right to vote on this ballot.
    /// May only be called by `chairperson`.
    function giveRightToVote(address toVoter) public {    21279 gas
        require(msg.sender == chairperson, "Only chairperson can give right to vote.");
        require(!voters[toVoter].voted, "The voter has already voted.");
        require(voters[toVoter].weight == 0, "The voter already has the right to vote.");

```

```
call to Ballot.proposals errored: execution reverted

call to Ballot.proposals

[call] from: 0x2986c1699AF0639a980e5eDD7EC5DedFeFCc89A1 to: Ballot.proposals(uint256) data: 0x013...00001
call to Ballot.voters

[call] from: 0x2986c1699AF0639a980e5eDD7EC5DedFeFCc89A1 to: Ballot.voters(address) data: 0xa3e...c89a1
call to Ballot.winningProposal

[call] from: 0x2986c1699AF0639a980e5eDD7EC5DedFeFCc89A1 to: Ballot.winningProposal() data: 0x609...ff1bd
```

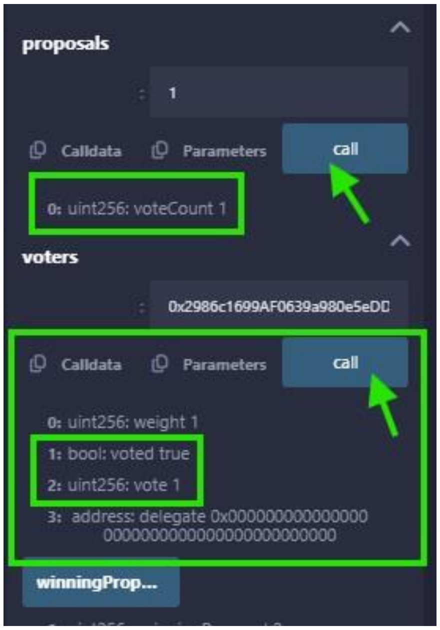
Account 1
0x2986c...c89A1

1.2282 SepoliaETH
+$0.00 (+0.00%) Portfolio

Nov 28, 2024

Give Right... Confirmed	-0 SepoliaETH -0 SepoliaETH
Give Right... Confirmed	-0 SepoliaETH -0 SepoliaETH
Give Right... Confirmed	-0 SepoliaETH -0 SepoliaETH

Now, we can see proposal index 1 (or proposal 2) has 1 vote, and Chairperson's status as a voter has been changed to **voted true**.



proposals

: 1

Calldata Parameters call

0: uint256: voteCount 1

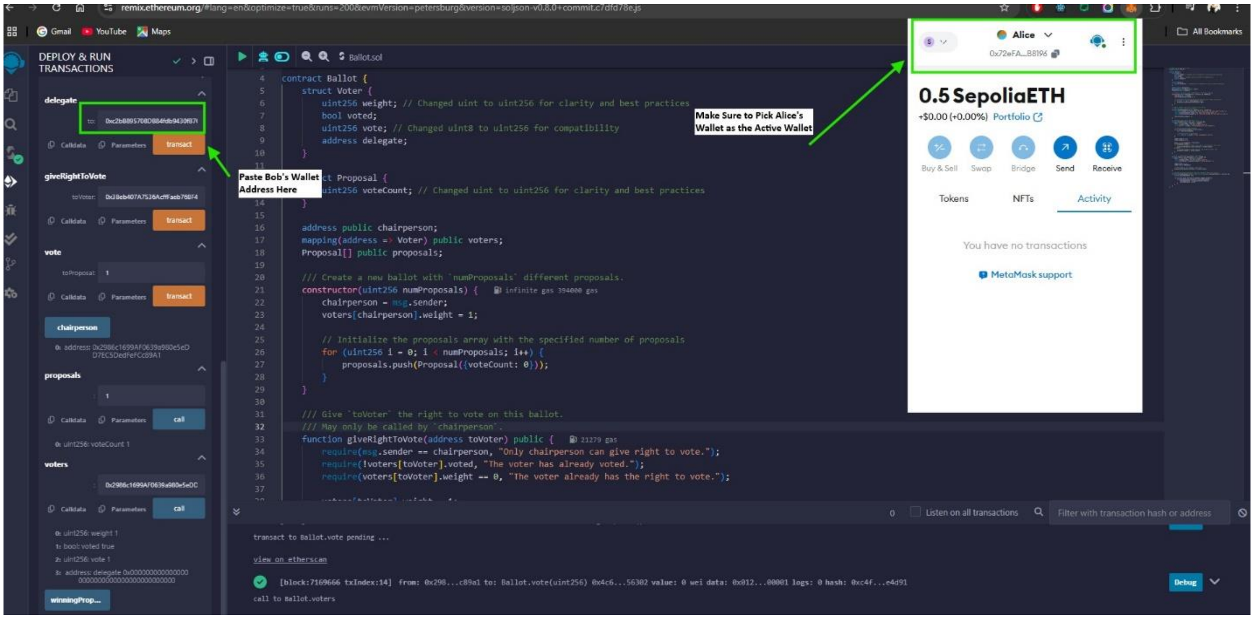
voters

: 0x2986c1699AF0639a980e5eDD

Calldata Parameters call

0: uint256: weight 1
1: bool: voted true
2: uint256: vote 1
3: address: delegate 0x000000000000000 0000000000000000000000000

Now, let's switch to Alice's account and delegate her vote to Bob by pasting Bob's address on delegate and click Transact while Alice's Wallet is active on MetaMask:



remix.ethereum.org/#lang=en&optimize=true&runs=200&evmVersion=petersburg&version=soljson-v0.8.0+commit.c7dfd78e.js
Gmail
YouTube
Maps
All Bookmarks
DEPLOY & RUN TRANSACTIONS
Ballot.sol

delegate

to: 0xc2b8895708D884fdb9430f87(

Calldata Parameters transact

giveRightToVote

toVoter: 0x38eb407A7536AcffFacb768F4

Calldata Parameters transact

vote

toProposal: 1

Calldata Parameters transact

chairperson

0: address: 0x2986c1699AF0639a980e5eD D7EC5DedFeFCc89A1

proposals

: 1

Calldata Parameters call

0: uint256: voteCount 1

voters

: 0x2986c1699AF0639a980e5eDC

Calldata Parameters call

0: uint256: weight 1
1: bool: voted true
2: uint256: vote 1
3: address: delegate 0x000000000000000 0000000000000000000000000

Paste Bob's Wallet Address Here

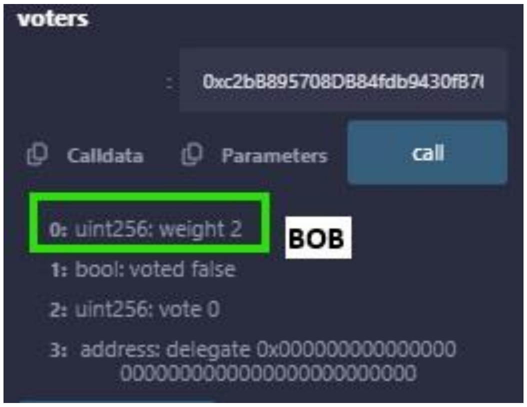
Make Sure to Pick Alice's Wallet as the Active Wallet

```
contract Ballot {
    struct Voter {
        uint256 weight; // Changed uint to uint256 for clarity and best practices
        bool voted;
        uint256 vote; // Changed uint8 to uint256 for compatibility
        address delegate;
    }

      ct Proposal {
        uint256 voteCount; // Changed uint to uint256 for clarity and best practices
    }

    address public chairperson;
    mapping(address => Voter) public voters;
    Proposal[] public proposals;

    /// Create a new ballot with `numProposals` different proposals.
    constructor(uint256 numProposals) {     infinite gas 394000 gas
        chairperson = msg.sender;
        voters[chairperson].weight = 1;

        // Initialize the proposals array with the specified number of proposals
        for (uint256 i = 0; i < numProposals; i++) {
            proposals.push(Proposal({voteCount: 0}));
        }
    }

    /// Give `toVoter` the right to vote on this ballot.
    /// May only be called by `chairperson`.
    function giveRightToVote(address toVoter) public {     21279 gas
        require(msg.sender == chairperson, "Only chairperson can give right to vote.");
        require(!voters[toVoter].voted, "The voter has already voted.");
        require(voters[toVoter].weight == 0, "The voter already has the right to vote.");
```

0
Listen on all transactions
Filter with transaction hash or address

```
transact to Ballot.vote pending ...

view on etherscan

[block:7169666 txIndex:14] from: 0x298...c89a1 to: Ballot.vote(uint256) 0x4c6...56302 value: 0 wei data: 0x012...00001 logs: 0 hash: 0xc4f...e4d91
call to Ballot.voters
```

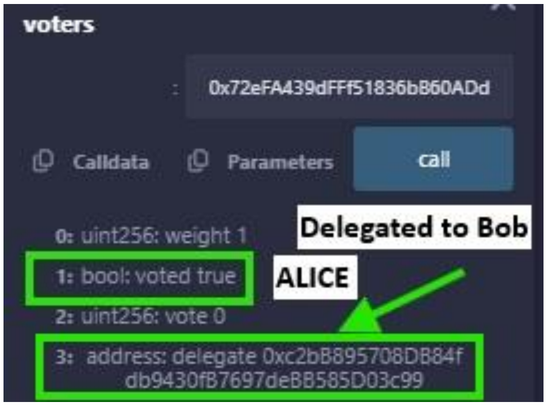
Now, let's check Bob's and Alice's status as a voter. Bob's vote has weight 2 now, and Alice's vote is considered as given, because she passed it to Bob. Now Bob will vote on behalf of Alice, and himself:



voters

: 0xc2bB895708DB84fdb9430fB7(

0: uint256: weight 2
1: bool: voted false
2: uint256: vote 0
3: address: delegate 0x0000000000000000000000000000000000000000

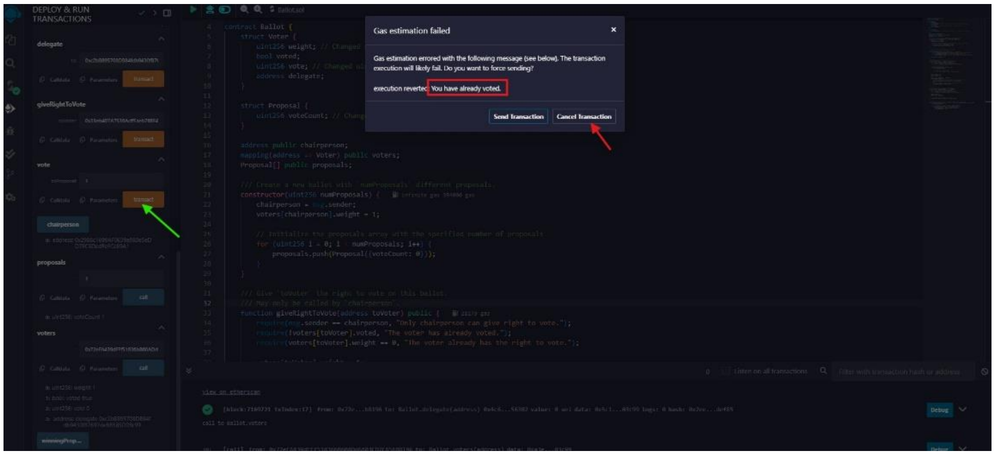


voters

: 0x72eFA439dFFf51836bB60ADd

0: uint256: weight 1
1: bool: voted true
2: uint256: vote 0
3: address: delegate 0xc2bB895708DB84fdb9430fB7697deBB585D03c99

Now, if we try to vote using Alice's account, it will throw an error saying Alice has already voted:

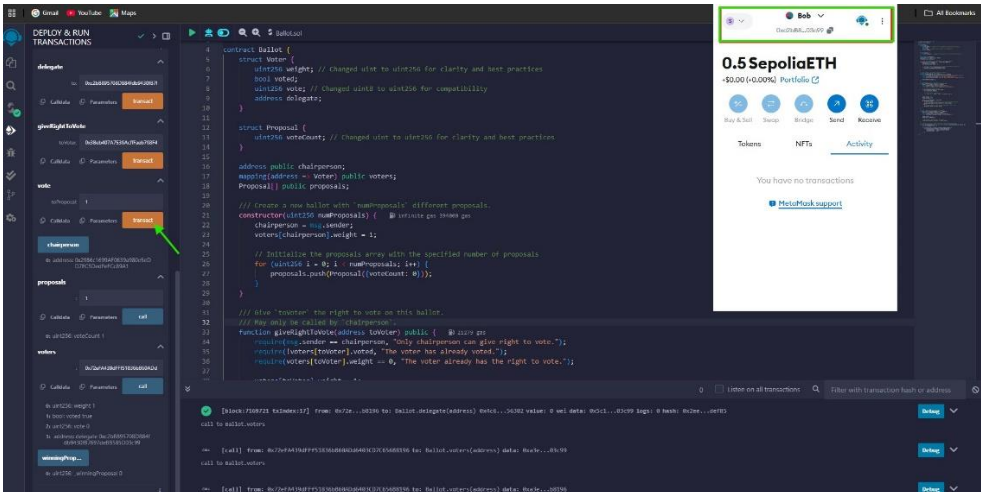


Gas estimation failed

Gas estimation errored with the following message (see below). The transaction execution will likely fail. Do you want to force sending?

execution reverted: You have already voted.

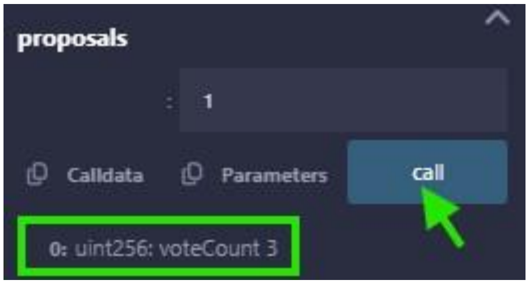
Now, let's vote for proposal index 1 (proposal 2) with Bob's account:



```
contract Ballot {
    struct Voter {
        uint256 weight; // Changed uint to uint256 for clarity and best practices
        bool voted;
        uint256 vote; // Changed uint8 to uint256 for compatibility
        address delegate;
    }

    struct Proposal {
        uint256 voteCount; // Changed uint to uint256 for clarity and best practices
    }

    address public chairperson;
    mapping(address => Voter) public voters;
    Proposal[] public proposals;

    /// Create a new ballot with `numProposals` different proposals.
    constructor(uint256 numProposals) {
        chairperson = msg.sender;
        voters[chairperson].weight = 1;

        // Initialize the proposals array with the specified number of proposals
        for (uint256 i = 0; i < numProposals; i++) {
            proposals.push(Proposal({voteCount: 0}));
        }
    }

    /// Give `toVoter` the right to vote on this ballot.
    /// May only be called by `chairperson`.
    function giveRightToVote(address toVoter) public {
        require(msg.sender == chairperson, "Only chairperson can give right to vote.");
        require(!voters[toVoter].voted, "The voter has already voted.");
        require(voters[toVoter].weight == 0, "The voter already has the right to vote.");
```

```
[block:7169721 txIndex:17] from: 0x72e...b8196 to: Ballot.delegate(address) 0x4c6...56302 value: 0 wei data: 0x5c1...03c99 logs: 0 hash: 0x2ee...def85
call to Ballot.voters
[call] from: 0x72eFA439dFFf51836bB60ADd6403CD7C65688196 to: Ballot.voters(address) data: 0xa3e...03c99
call to Ballot.voters
[call] from: 0x72eFA439dFFf51836bB60ADd6403CD7C65688196 to: Ballot.voters(address) data: 0xa3e...b8196
```

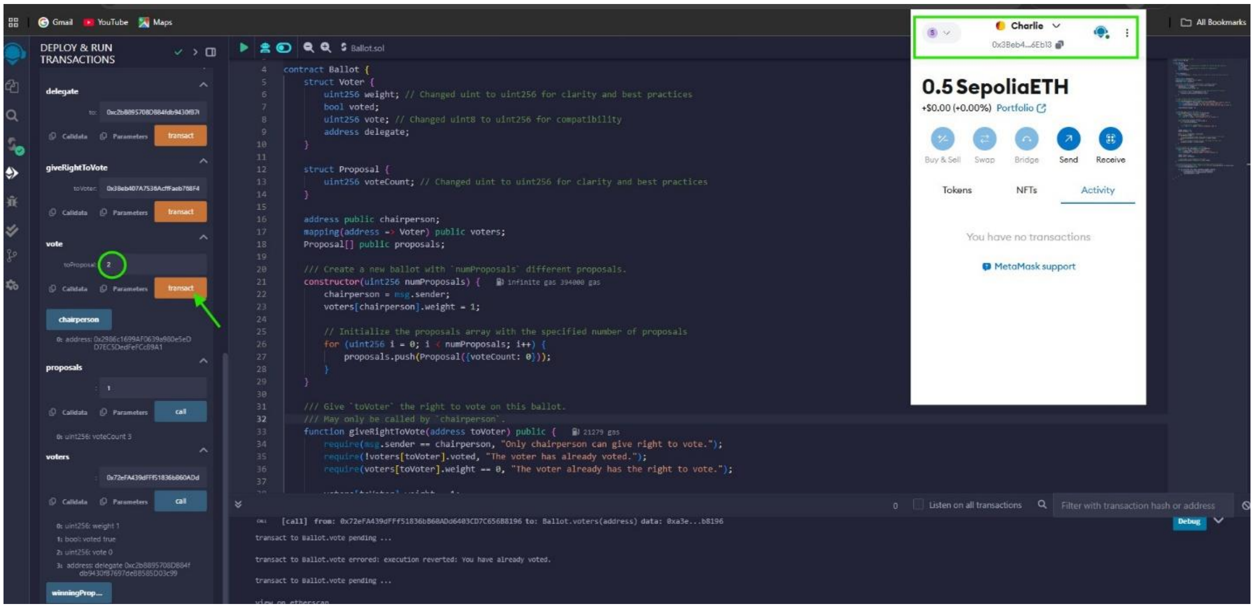
Now, Proposal index 1 has 3 votes in total:



: 1

0: uint256: voteCount 3

Let’s vote for Proposal index 2 (Proposal 3) with Charlie’s Account:



delegate
to: 0xc2b8895708D884fdb9430f87I

giveRightToVote
toVoter: 0x38eb407A7538AcffFaeb788F4

vote
toProposal: 2

chairperson
0: address: 0x2986c1699AF0639a980e5eD D7EC5DedFeFCc89A1

proposals
: 1
0: uint256: voteCount 3

voters
: 0x72eFA439dFFf51836b860ADd
0: uint256: weight 1
1: bool: voted true
2: uint256: vote 0
3: address: delegate 0xc2b8895708D884f db9430f87697deB8585D03c99

```
contract Ballot {
    struct Voter {
        uint256 weight; // Changed uint to uint256 for clarity and best practices
        bool voted;
        uint256 vote; // Changed uint8 to uint256 for compatibility
        address delegate;
    }

    struct Proposal {
        uint256 voteCount; // Changed uint to uint256 for clarity and best practices
    }

    address public chairperson;
    mapping(address => Voter) public voters;
    Proposal[] public proposals;

    /// Create a new ballot with `numProposals` different proposals.
    constructor(uint256 numProposals) {     infinite gas 394000 gas
        chairperson = msg.sender;
        voters[chairperson].weight = 1;

        // Initialize the proposals array with the specified number of proposals
        for (uint256 i = 0; i < numProposals; i++) {
            proposals.push(Proposal({voteCount: 0}));
        }
    }

    /// Give `toVoter` the right to vote on this ballot.
    /// May only be called by `chairperson`.
    function giveRightToVote(address toVoter) public {     21279 gas
        require(msg.sender == chairperson, "Only chairperson can give right to vote.");
        require(!voters[toVoter].voted, "The voter has already voted.");
        require(voters[toVoter].weight == 0, "The voter already has the right to vote.");
```

0 ☐ Listen on all transactions Filter with transaction hash or address

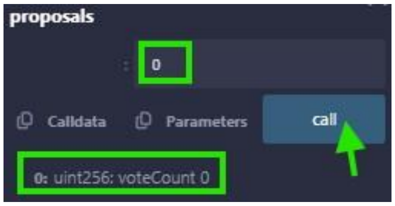
```
[call] from: 0x72eFA439dFFf51836bB60ADd6403CD7C656B8196 to: Ballot.voters(address) data: 0xa3e...b8196
transact to Ballot.vote pending ...

transact to Ballot.vote errored: execution reverted: You have already voted.

transact to Ballot.vote pending ...

view on etherscan
```

Charlie
0x3Beb4...6Eb13

0.5 SepoliaETH
+$0.00 (+0.00%) Portfolio

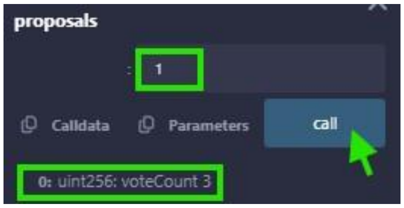
You have no transactions

Now, Proposal index 0 has 0 vote, 1 has 3 votes, and 2 has 1 vote:



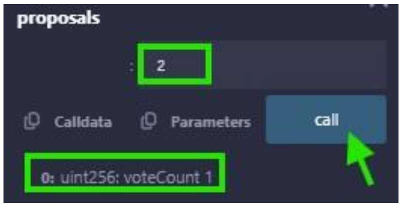
: 0

0: uint256: voteCount 0



: 1

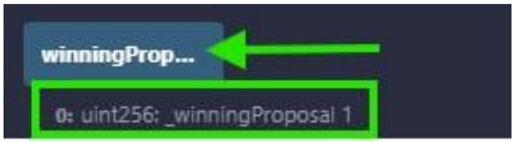
0: uint256: voteCount 3



: 2

0: uint256: voteCount 1

So, the winning proposal is the proposal index 1, or the 2nd proposal:



0: uint256: _winningProposal 1

GitHub Repository: https://github.com/Salekin-Nabil/Blockchain-Based-Ballot/tree/main